# Appendix D

# Module Internal Designs

## D.1 BSHAM Modules

### D.1.1 *absmach* MID

**state variables**

```
int acc,pc;
int mem[AM_MEMSIZ];
```

**state invariant**

1. $\texttt{acc} \in [0..\texttt{AM\_MAXINT}]$
2. $\texttt{pc} \in [0..\texttt{AM\_MEMSIZ} - 1]$
3. $(\forall i \in [0..\texttt{AM\_MEMSIZ} - 1])(\texttt{mem}[i] \in [0..\texttt{AM\_MAXINT}])$

**considerations**

Since the abstract and concrete states are identical, the abstraction function and the access routine semantics are omitted.

### D.1.2 *exec* MID

There is no MID for *exec*.

### D.1.3 *load* MID

There is no MID for *load*.

### D.1.4 *sham* MID

There is no MID for *sham*.

### D.1.5 *token* MID

**state variables**
`char buf[TK_MAXSTRLEN+2];`
`int cur;`

**state invariant**
1. $\texttt{buf}[0..\texttt{TK\_MAXSTRLEN}+1]$ contains a null.
2. $\texttt{cur} \in [0..\mathit{leftnull}(\texttt{buf})]$
3. $\texttt{cur} < \mathit{leftnull}(\texttt{buf}) \rightarrow (\texttt{buf}[\texttt{cur}] \neq \texttt{'~~'} \wedge \texttt{buf}[\mathit{leftnull}(\texttt{buf}) - 1] = \texttt{'~~'})$

**abstraction function**
$\mathit{toklist}$ = the sequence of tokens in $\texttt{buf}[\texttt{cur}..\mathit{leftnull}(\texttt{buf}) - 1]$

**access routine semantics**
`tk_s_init`:
  **transition**: $\texttt{buf}, \texttt{cur} := \texttt{""}, 0$
  **exceptions**: none
$\texttt{tk\_s\_str}(s)$:
  **transition**:
    $\texttt{buf} :=$ (there is a token in $s[0..\mathit{leftnull}(s) - 1] \Rightarrow \mathit{rmblanks}(s) \parallel \texttt{"~~"}$
      $\mid \mathit{true} \Rightarrow \texttt{""}$)
    $\texttt{cur} := 0$
  **exceptions**: $\mathit{exc} := (|s| > \texttt{TK\_MAXSTRLEN} \Rightarrow \texttt{tk\_maxlen})$
`tk_sg_next`:
  **transition/output**:
    Let $\mathit{curtok}$ be the token beginning at $\texttt{buf}[\texttt{cur}]$
      $\mathit{curtoktyp}$ be the token type of $\mathit{curtok}$
    $\mathit{out} := \langle \mathit{curtok}, \mathit{curtoktyp} \rangle$
    $\texttt{cur} :=$ (there is a token, beginning at position $i$,
      in $\texttt{buf}[\texttt{cur} + |\mathit{curtok}| + 1..\mathit{leftnull}(\texttt{buf}) - 1]$
        $\Rightarrow i$
      $\mid \mathit{true} \Rightarrow \mathit{leftnull}(\texttt{buf})$)
  **exceptions**: $\mathit{exc} := (\texttt{buf}[\texttt{cur}] = \mathit{null} \Rightarrow \texttt{tk\_end})$
`tk_g_end`:
  **output**: $\mathit{out} := (\texttt{buf}[\texttt{cur}] = \text{null})$
  **exceptions**: none

**local functions**
$\mathit{leftnull} : \mathit{string} \rightarrow \mathit{integer}$
  $\mathit{leftnull}(s) :=$ (there is a null in $s \Rightarrow$ the index of the leftmost one)
$\mathit{rmblanks} : \mathit{string} \rightarrow \mathit{string}$
  $\mathit{rmblanks}(s) := s$, with leading blanks removed

## D.2 ISHAM Modules

### D.2.1 *keybdin* MID

There is no MID for *keybdin*.

### D.2.2 *scndr* MID

There is no MID for *scndr*.

### D.2.3 *scngeom* MID

There is no MID for *scngeom*.

### D.2.4 *scnstr* MID

There is no MID for *scnstr*.

## D.3 Demonstration Modules

### D.3.1 *stack* MID

**state variables**
`int stack[PS_MAXSIZ];`
`int siz;`

**state invariant**
$\mathtt{siz} \in [0..\mathtt{PS\_MAXSIZ}]$

**abstraction function**
$|s| = \mathtt{siz} \wedge (\forall i \in [0..\mathtt{siz}-1])(s[i] = \mathtt{stack}[i])$

**access routine semantics**
`ps_s_init`:
  **transition**: $\mathtt{siz} := 0$
  **exceptions**: none
`ps_s_push`($x$):
  **transition**: $\mathtt{stack}[\mathtt{siz}], \mathtt{siz} := x, \mathtt{siz}+1$
  **exceptions**: $exc := (\mathtt{siz} = \mathtt{PS\_MAXSIZ} \Rightarrow \mathtt{ps\_full})$
`ps_s_pop`:
  **transition**: $\mathtt{siz} := \mathtt{siz} - 1$
  **exceptions**: $exc := (\mathtt{siz} = 0 \Rightarrow \mathtt{ps\_empty})$
`ps_g_top`:
  **output**: $out := \mathtt{stack}[\mathtt{siz} - 1]$
  **exceptions**: $exc := (\mathtt{siz} = 0 \Rightarrow \mathtt{ps\_empty})$

ps_g_depth:
    **output**: $out$ := siz
    **exceptions**: none

### D.3.2 *symtbl* MID

**state variables**

```
struct {
    char sym[ST_MAXSYMLEN+1];
    int loc;
} tbl[ST_MAXSYMS];
int tblcnt;
```

**state invariant**

1. Every symbol in tbl$[0..$tblcnt$-1]$ contains a null.
2. There are no duplicate symbols in tbl$[0..$tblcnt$-1]$.
3. tblcnt $\in [0..$ST_MAXSYMS$]$

**abstraction function**

$tbl = \{\langle sym, loc\rangle \mid (\exists i \in [0..\texttt{tblcnt}-1])(sym = \texttt{tbl}[i].\texttt{sym} \wedge loc = \texttt{tbl}[i].\texttt{loc})\}$

**access routine semantics**

st_s_init:
    **transition**: tblcnt := 0
    **exceptions**: none

st_s_add($sym, loc$):
    **transition**: tblcnt, tbl[tblcnt] := tblcnt + 1, $\langle sym, loc\rangle$
    **exceptions**: $exc$ := $(|sym| > \texttt{ST\_MAXSYMLEN} \Rightarrow \texttt{st\_maxlen}$
    $\mid findsym(sym) \neq \texttt{NOTFOUND} \Rightarrow \texttt{st\_exsym}$
    $\mid \texttt{tblcnt} = \texttt{ST\_MAXSYMS} \Rightarrow \texttt{st\_full})$

st_g_exsym($sym$):
    **output**: $out := (findsym(sym) \neq \texttt{NOTFOUND})$
    **exceptions**: none

st_s_loc($sym, loc$):
    **transition**: tbl[$findsym(sym)$].loc := loc
    **exceptions**: $exc := (findsym(sym) = \texttt{NOTFOUND} \Rightarrow \texttt{st\_notexsym})$

st_g_loc($sym$):
    **output**: $out$ := tbl[$findsym(sym)$].loc
    **exceptions**: $exc := (findsym(sym) = \texttt{NOTFOUND} \Rightarrow \texttt{st\_notexsym})$

st_g_siz:
    **output**: $out$ := tblcnt
    **exceptions**: none

**local constants**

`#define NOTFOUND -1`

**local functions**

$findsym : string \rightarrow integer$

$$findsym(s) = \quad ((\exists i \in [0..\texttt{tblcnt} - 1])(s = \texttt{tbl}[i].\texttt{sym}) \Rightarrow i$$
$$\mid true \Rightarrow \texttt{NOTFOUND})$$